and eleventh bit of sin(4 π / 64) **58** and twelfth bit of sin(4 π / 64) **60** are pairs of adjacent bits having value 1.

The first partial product computed for the product of the 16-bit twos complement representation of sin(2 π / 64) and the arbitrary number could be shifted eleven places to obtain the contribution of fourth bit of sin(4 π / 64) **38** and fifth bit of sin(4 π / 64) **40** to the product of the 16-bit twos complement representation of sin(4 π / 64) and the arbitrary number. Likewise, the first partial product could be shifted six places or four places to obtain the contributions of ninth bit of sin(4 π / 64) **54** and tenth bit of sin(4 π / 64) **56,** and the contributions of eleventh bit of sin(4 π / 64) **58** and twelfth bit of sin(4 π / 64) **60** respectively.

In this simple example, the 16-bit twos complement representation of sin(2 π / 64) corresponds to the second real number of claim **1**, 16-bit twos complement representation of sin(4 π / 64) corresponds to the third real number, and the arbitrary number corresponds to the first real number. The first partial product is an intermediate term computed by a multiplier calculating the product of the first real number and the second real number. The presence of adjacent pairs of bits with value 1 in the representations of both sin(2 π / 64) and sin(4 π / 64) means that the first partial product does not have to be re-calculated in computing the second product. It can be shared.

Note that the possibility of sharing the first partial product in the two product computations is not apparent from 16-bit decimal value of sin(2 π / 64) **62** to six decimal places and 16-bit decimal value of sin(4 π / 64) **64** to six decimal places, or from desired decimal value of sin(2 π / 64) **46** to six places and desired decimal value of sin(4 π / 64) **48** to six decimal places. Sharing of the first partial product depends on the representations of the second real number and the third real number, not just on their number values. Alternative representations of the same number values in other finite-precision numeric formats, such as signed integer or signed decimal, may offer different opportunities for shared computation.

CONCLUSION, RAMIFICATIONS, AND SCOPE

The reader will see that the present invention has several advantages over prior art techniques for multiplication of number pairs, particularly in signal processing transforms which use sums of products. These signal processing transforms often have outputs that are sums of weighted inputs. The weights may be fixed and known in advance, or may take on a restricted set of possible values. Likewise, in some applications the inputs may come from a limited set of possible values. The invention exploits restrictions on the relationship between number values, representations, or number values and representations of two number pairs being multiplied in order to reduce the cost of implementing the multiplication operations.

In a preferred embodiment of the invention, a first multiplier computes a first product which is the product of a first real number and a second real number, as well as a first set of intermediate terms. Rather than performing an entirely independent computation, a second multiplier computes the product of the first real number and a third real number using one or more members of a set consisting of the first set of intermediate terms computed by the first multiplier and the first product. The two multipliers can exploit the relationship between a representation or number value of the second real number and a representation or number value of the third real number to decrease the cost of the computation.

In an alternative embodiment of the invention, the second multiplier is not able to compute the product of the first real number and the second real number. In another alternative embodiment of the invention, the first multiplier is not able to compute the product of the first real number and the third real number. In these embodiments, reduced-complexity multipliers such as constant multipliers or the non-general, non-constant multipliers discussed in the application NON-CONSTANT REDUCED-COMPLEXITY MULTIPLICATION FOR SIGNAL PROCESSING TRANSFORMS by the inventor of the present invention can be used.

In an alternative embodiment of the invention, the second multiplier uses members of the set of intermediate terms generated by the first multiplier, but not the first product. This embodiment allows for parallel implementation of the first multiplier and the second multiplier.

The invention is used in computing sums of products. It is particularly useful in transforms with fixed weights, especially when the transform is used repeatedly in a signal processing application. The numbers can be Cartesian components of complex numbers. Alternative embodiments of the invention include a number to be multiplied that is a Cartesian component of a discrete Fourier transform weight or of a discrete Fourier transform input. Alternative embodiments of the invention include a number to be multiplied that is a Cartesian component of an inverse discrete Fourier transform weight or of an inverse discrete Fourier transform input. Still another alternative embodiment of the invention includes restrictions on the allowed values of both numbers being multiplied, which may enable even greater reduction of computational complexity.

The invention can be used in computing discrete cosine transforms, discrete sine transforms, inverse discrete cosine transforms, inverse discrete sine transforms, and other transforms. The invention can be used for digital filtering. The invention can be used for pulse-shaping in digital communications, or for digital modulation.

The invention is not limited to particular number representations or to particular applications. Signal processing transforms that use multiple sums of products are used in digital communications, radar, sonar, astronomy, geology, control systems, image processing, and video processing. Technologies used to implement signal processing transforms include hardware technologies such as application specific integrated circuits and field-programmable gate arrays and software technologies such as multiplication on a general-purpose microprocessor.